

FIG. 1

14 {
 14a {
 * 1. productive class:
 * definition
 class OPERATIONS definition.
 public section.
 class-methods:
 ADD importing A type I
 B type I
 returning VALUE(RESULT) type I.,
 endclass.
 14b {
 * implementation
 class OPERATIONS implementation.
 method ADD.
 RESULT = A + B.
 endmethod.
 endclass.

16 {
 18a {
 * 2. test class:
 * definition
 class TEST_OPERATIONS definition for testing.
 public section.
 methods TEST_ADD (for testing)
 endclass.
 18b {
 * implementation
 class TEST_OPERATIONS implementation.
 method TEST_ADD.
 * test data: variable needed to store the result from the productive method:
 data: ACTUAL_RESULT type I.
 * call the method under test:
 ACTUAL_RESULT = OPERATIONS-ADD(A = 3 B = 5).
 * compare the result with the expected value:
 CL_AUNIT_ASSERT->ASSERT_EQUALS(
 ACT = ACTUAL_RESULT
 EXP = 8
 MSG = 'this is the message which occurs if the test failed'
).
 endmethod.
 endclass.

F16.2

80

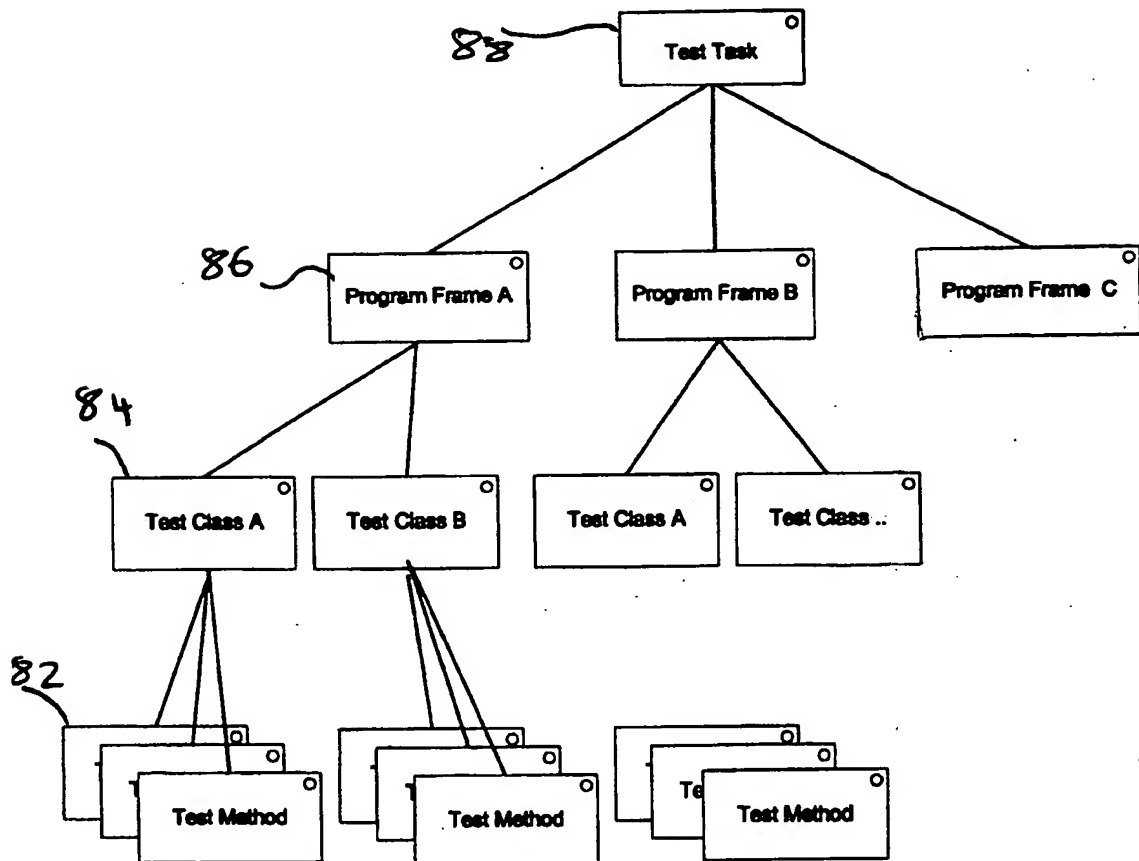
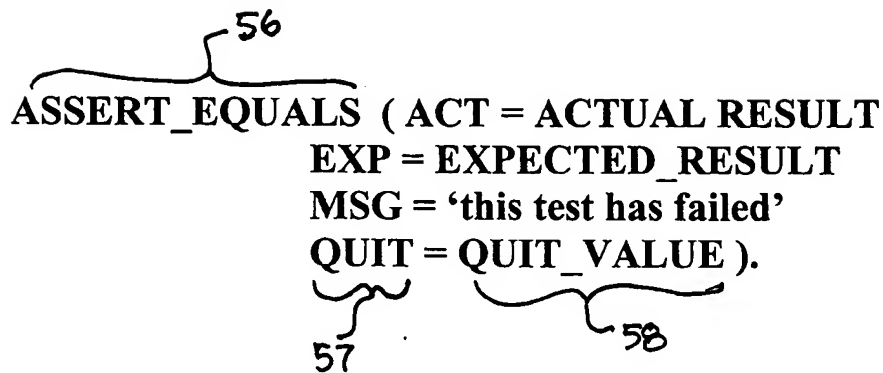


FIG. 3


The diagram shows the function signature **ASSERT_EQUALS (ACT = ACTUAL RESULT
EXP = EXPECTED_RESULT
MSG = 'this test has failed'
QUIT = QUIT_VALUE).** with three hand-drawn curly braces and numbers: a brace above the opening parenthesis labeled '56', a brace under the **QUIT** parameter labeled '57', and a brace under the **QUIT_VALUE** parameter labeled '58'.

Where **QUIT_VALUE** defines at which level the test flow should be interrupted:

- **NO**: continue the current test method.
- **METHOD**: interrupt the current test method.
- **CLASS**: interrupt the test class execution.
- **PROGRAM**: abandon all test class executions of the currently tested program frame.

FIG. 4

100

6

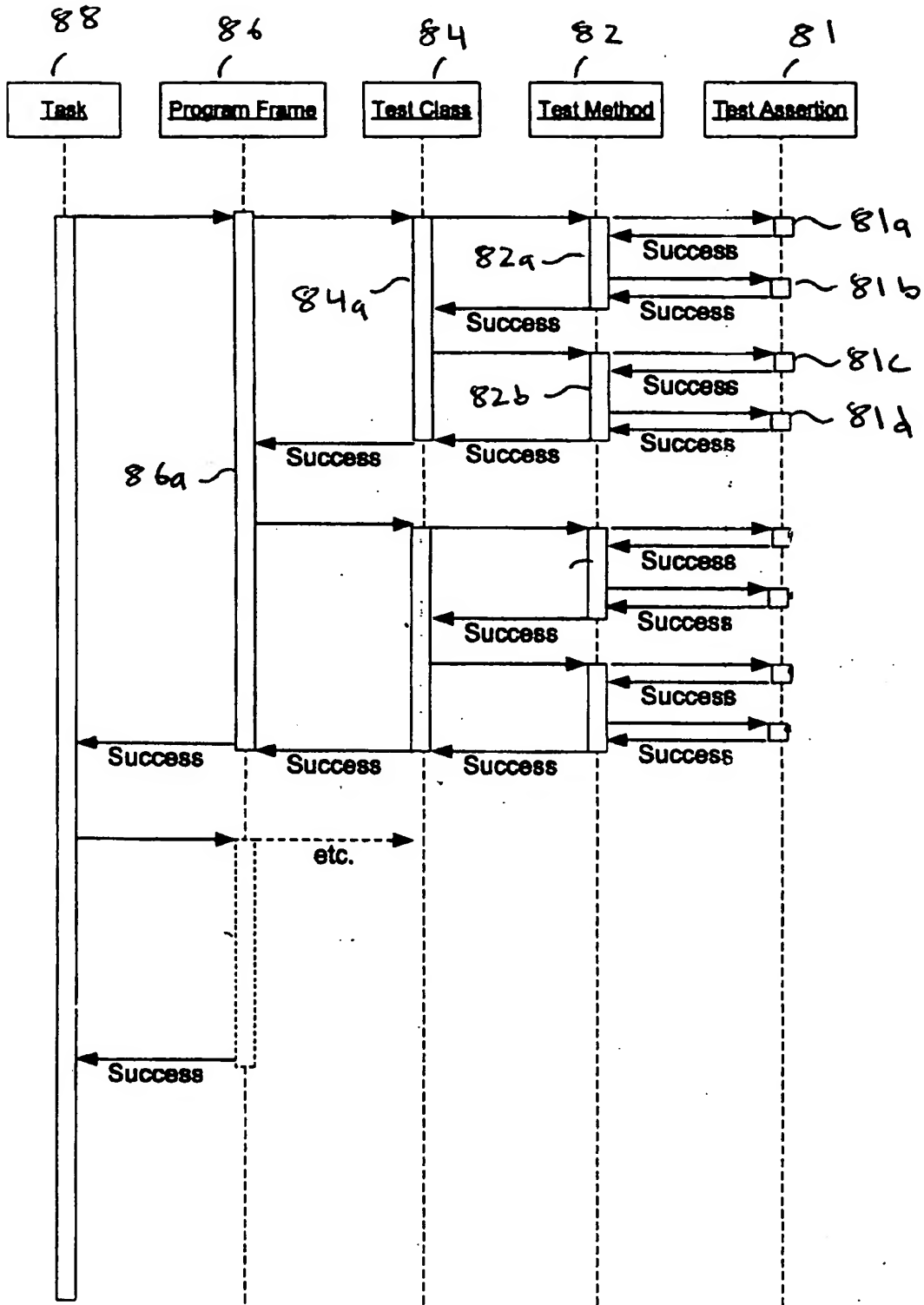


FIG. 5

200

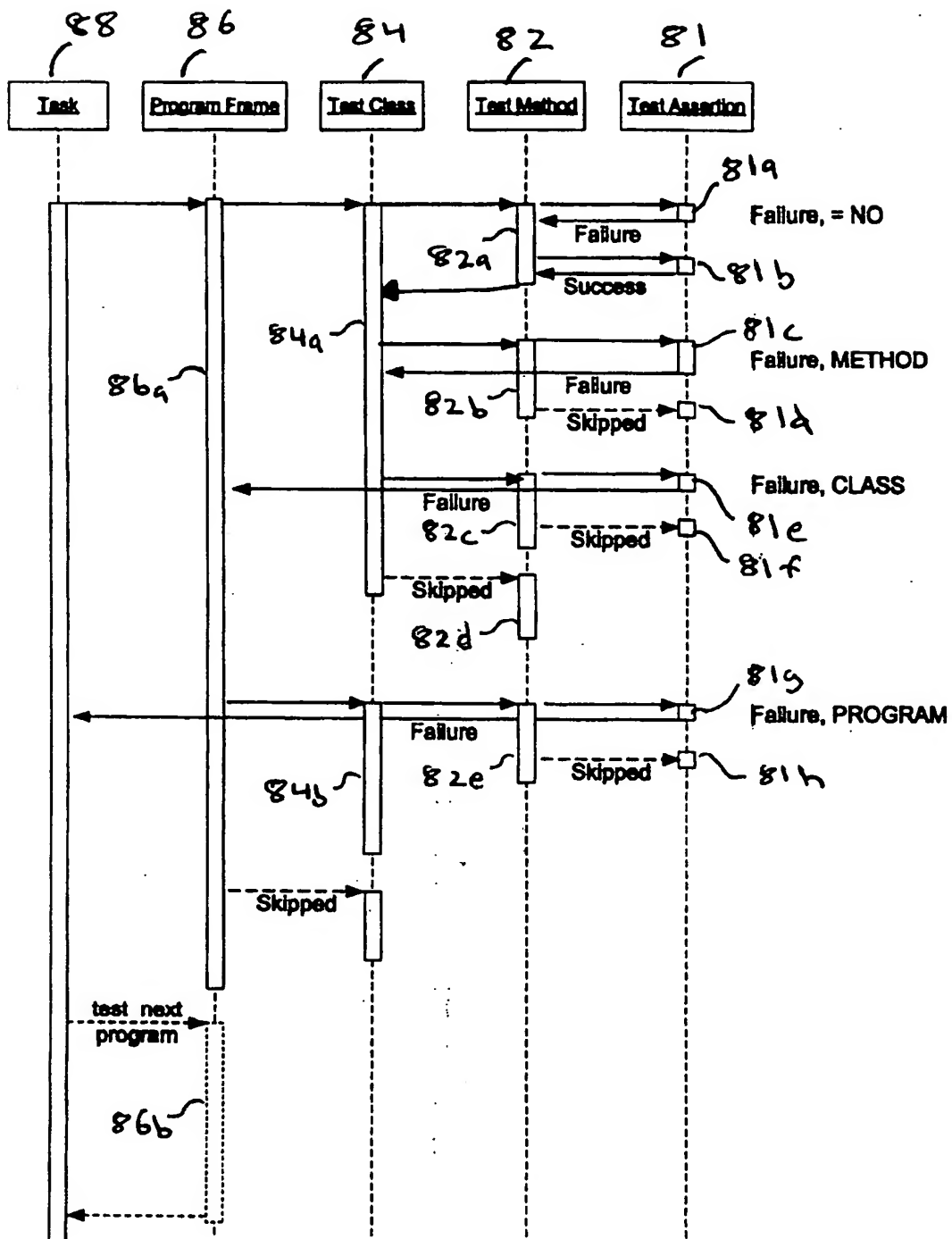


FIG. 6